

Adaptive Gaussian Mixture Filter Based on Statistical Linearization

Marco F. Huber

Variable Image Acquisition and Processing Research Group
Fraunhofer Institute of Optronics, System Technologies and Image Exploitation IOSB
Karlsruhe, Germany
Email: marco.huber@ieee.org

Abstract—Gaussian mixtures are a common density representation in nonlinear, non-Gaussian Bayesian state estimation. Selecting an appropriate number of Gaussian components, however, is difficult as one has to trade off computational complexity against estimation accuracy. In this paper, an adaptive Gaussian mixture filter based on statistical linearization is proposed. Depending on the nonlinearity of the considered estimation problem, this filter dynamically increases the number of components via splitting. For this purpose, a measure is introduced that allows for quantifying the locally induced linearization error at each Gaussian mixture component. The deviation between the nonlinear and the linearized state space model is evaluated for determining the splitting direction. The proposed approach is not restricted to a specific statistical linearization method. Simulations show the superior estimation performance compared to related approaches and common filtering algorithms.

Keywords: Bayesian estimation, nonlinear filtering, statistical linearization, Kalman filtering, Gaussian mixtures.

I. INTRODUCTION

Bayesian state estimation for nonlinear systems requires an efficient approximation for practical applications as closed-form solutions are not available in general. A common approximation technique is the discretization of the state space as done in grid filters or particle filters [1]. Theoretically, these techniques facilitate to approach the true statistics of the state with arbitrary accuracy. But they are only applicable to low-dimensional problems since their computational complexity increases exponentially with the dimension of the state space.

A famous exception that exhibits an analytic solution is the linear Gaussian case. Here, the famous Kalman filter provides optimal results in an efficient manner [2]. So-called Gaussian filters try to adapt the Kalman filter equations to nonlinear problems by assuming that the density function of the state can be represented by a Gaussian density. The extended Kalman filter [3] applies first-order Taylor series expansion for linearization. The unscented Kalman filter [4], [5] or the Gaussian estimator [6] offer higher order accuracy by employing statistical linearization. But in general a single Gaussian density is typically not a sufficient representation for the true density function, which may be skew or multimodal. Thanks to their universal approximator property, Gaussian mixtures [7] are a much better approach for approximating complex density functions. Examples for Gaussian mixture filters applied to nonlinear estimation are in [8], [9].

The estimation accuracy of Gaussian mixture filters significantly depend on the number of Gaussian components used. This number is typically defined by the user. In this paper, a novel Gaussian mixture filter is proposed, which adapts the number of components dynamically and on-line. The nonlinear system and measurement models are linearized locally by means of statistical linearization at each component of the Gaussian mixture. The induced linearization error is quantified by means of the linearization error covariance matrix. Based on this error, a novel moment-preserving splitting procedure is proposed for introducing new mixture components. The component causing the highest linearization error is selected, while splitting is performed in direction of the strongest nonlinearity, i.e., the strongest deviation between the nonlinear model and its linearized version. Both linearization and splitting are independent of the used statistical linearization method, which makes the proposed filter versatily applicable.

The paper is structured as follows: The Bayesian state estimation problem is formulated in the next section. In Sec. III, a brief introduction in statistical linearization is given. The novel splitting scheme is derived in Sec. IV. Based on this, Sec. V describes the complete adaptive Gaussian mixture filter with all major components. Numerical evaluation by means of simulations is part of Sec. VI. The paper closes with concluding remarks.

II. PROBLEM FORMULATION

In this paper, discrete-time nonlinear dynamic systems

$$\underline{\mathbf{x}}_{k+1} = \underline{\mathbf{a}}_k(\underline{\mathbf{x}}_k, \underline{\mathbf{u}}_k, \underline{\mathbf{w}}_k), \quad (1)$$

$$\underline{\mathbf{z}}_k = \underline{\mathbf{h}}_k(\underline{\mathbf{x}}_k, \underline{\mathbf{v}}_k) \quad (2)$$

are considered. Here, (1) is the dynamics model with the known time-variant nonlinear system function $\underline{\mathbf{a}}_k(\cdot)$, which propagates the system state¹ $\underline{\mathbf{x}}_k \in \mathbb{R}^{n_x}$ at time step k to time step $k + 1$, given the current system input $\underline{\mathbf{u}}_k \in \mathbb{R}^{n_u}$ and the process noise $\underline{\mathbf{w}}_k \in \mathbb{R}^{n_w}$. The measurement model is given by (2), where $\underline{\mathbf{h}}_k(\cdot)$ is the known time-variant nonlinear measurement function, $\underline{\mathbf{z}}_k \in \mathbb{R}^{n_z}$ is the measurement vector, and $\underline{\mathbf{v}}_k \in \mathbb{R}^{n_v}$ is the measurement noise. Note that an actual measurement value $\underline{\mathbf{z}}_k$ is a realization of the random vector $\underline{\mathbf{z}}_k$ in (2).

¹Random vectors are denoted by boldface letters.

Both noise processes \underline{w}_k and \underline{v}_k are assumed to be independent and white. The probability density functions of \underline{w}_k and \underline{v}_k are denoted by $f_k^w(\underline{w}_k)$ and $f_k^v(\underline{v}_k)$, respectively. It is assumed that these density functions are described via *Gaussian mixtures*

$$f_k^w(\underline{w}_k) = \sum_{i=1}^{L_k^w} \omega_{k,i}^w \cdot \mathcal{N}(\underline{w}_k; \hat{\underline{w}}_{k,i}, \mathbf{C}_{k,i}^w), \quad (3)$$

$$f_k^v(\underline{v}_k) = \sum_{i=1}^{L_k^v} \omega_{k,i}^v \cdot \mathcal{N}(\underline{v}_k; \hat{\underline{v}}_{k,i}, \mathbf{C}_{k,i}^v), \quad (4)$$

where L_k^w, L_k^v are the numbers of mixture components, $\omega_{k,i}^w, \omega_{k,i}^v$ are non-negative weights that sum up to one, and $\mathcal{N}(\underline{w}; \hat{\underline{w}}, \mathbf{C}^w)$ is a Gaussian density with mean vector $\hat{\underline{w}}$ and covariance matrix \mathbf{C}^w . The initial density function $f_0^x(\underline{x}_0)$ of the system state at time step $k = 0$ is also assumed to be given as a Gaussian mixture.

Estimating the system state from noisy measurements is done according to the Bayesian framework. Here, two steps are performed alternately, namely the prediction step and the filtering step. In the prediction step, the density $f_k^x(\underline{x}_k) := f_k^x(\underline{x}_k | \underline{u}_{0:k}, \underline{z}_{0:k})$ of the previous filtering step is propagated to the next time step according to

$$\begin{aligned} f_{k+1}^p(\underline{x}_{k+1}) &:= f_{k+1}^x(\underline{x}_{k+1} | \underline{u}_{0:k}, \underline{z}_{0:k}) \\ &= \int \underbrace{f(\underline{x}_{k+1} | \underline{x}_k, \underline{u}_k, \underline{w}_k)}_{\delta(\underline{x}_{k+1} - \underline{u}_k(\underline{x}_k, \underline{u}_k, \underline{w}_k))} \cdot f_k^e(\underline{x}_k) \cdot f_k^w(\underline{w}_k) d\underline{x}_k d\underline{w}_k, \end{aligned} \quad (5)$$

where $\underline{z}_{0:k} = (\underline{z}_0, \underline{z}_1, \dots, \underline{z}_k)$ denotes the measurements up to and including time step k , $f(\underline{x}_{k+1} | \underline{x}_k, \underline{u}_k, \underline{w}_k)$ is the transition density depending on the dynamics model (1), and $\delta(\cdot)$ is the Dirac delta distribution.

The filtering step determines the *posterior density* $f_k^e(\underline{x}_k)$ of the system state \underline{x}_k based on all acquired measurement values according to *Bayes' law*

$$f_k^e(\underline{x}_k) = c_k \cdot f(\underline{z}_k | \underline{x}_k) \cdot f_k^p(\underline{x}_k),$$

where c_k is a normalization constant and $f(\underline{z}_k | \underline{x}_k)$ is the likelihood function given by

$$f(\underline{z}_k | \underline{x}_k) = \int \delta(\underline{z}_k - \underline{h}_k(\underline{x}_k, \underline{v}_k)) \cdot f_k^v(\underline{v}_k) d\underline{v}_k$$

and the measurement model (2).

In general, for arbitrary nonlinear systems with arbitrarily distributed random vectors, there exist no analytical solutions of the prediction step and filtering step. Thus, for efficient estimation, it is inevitable to apply an approximate solution. In the following, an adaptive approximation scheme is proposed, where the predicted and posterior state densities are represented by means of Gaussian mixtures

$$f_k^\bullet(\underline{x}_k) = \sum_{i=1}^{L_k^\bullet} \omega_{k,i}^\bullet \cdot \mathcal{N}(\underline{x}_k; \hat{\underline{x}}_{k,i}, \mathbf{C}_{k,i}^\bullet), \quad \bullet \in \{e, p\} \quad (6)$$

where the number L_k^\bullet of mixture components is variable and adapted on-line by the proposed Gaussian mixture filter.

III. STATISTICAL LINEARIZATION

Substituting the Gaussian mixtures representing the noise and the state density into the prediction step and the filtering step, it can be easily seen that estimation can be performed component-wise. For example in case of the prediction, using (3) and (6) with $\bullet = e$ in (5) gives rise to

$$\begin{aligned} f_{k+1}^p(\underline{x}_{k+1}) &= \sum_{i=1}^{L_k^e} \sum_{j=1}^{L_k^w} \omega_{k,i}^e \cdot \omega_{k,j}^w \cdot \left(\int f(\underline{x}_{k+1} | \underline{x}_k, \underline{u}_k, \underline{w}_k) \cdot \right. \\ &\quad \left. \mathcal{N}(\underline{x}_k; \hat{\underline{x}}_{k,i}^e, \mathbf{C}_{k,i}^e) \cdot \mathcal{N}(\underline{w}_k; \hat{\underline{w}}_{k,j}^w, \mathbf{C}_{k,j}^w) d\underline{x}_k d\underline{w}_k \right). \end{aligned} \quad (7)$$

Thus, it is sufficient to focus in the following on the simplified nonlinear transformation

$$\underline{y} = \underline{g}(\underline{x}), \quad (8)$$

which maps the Gaussian random vector \underline{x} with density $\mathcal{N}(\underline{x}; \hat{\underline{x}}, \mathbf{C}^x)$ to the random vector \underline{y} . This nonlinear transformation can be replaced by $\underline{a}_k(\cdot)$ in the prediction step and by $\underline{h}_k(\cdot)$ in the filtering step, while the Gaussian random vector \underline{x} in (8) represents the joint Gaussian of the state and noise.

A. Classical Linearization

Calculating the density or the statistics of \underline{y} cannot be carried out in closed form. Hence, *directly* processing the density or the moments is computationally demanding and imprecise, or even impossible. An exception are linear transformations, where the Kalman filter [2] provides analytic expressions of the Bayesian estimation problem. To apply the Kalman filter equations to nonlinear transformations, a typical way is to linearize the nonlinear transformation, which results in the extended Kalman filter [3]. Here, it is assumed that the nonlinear transformation can be approximated by a linear transformation through a first-order Taylor series expansion around the mean $\hat{\underline{x}}$. In case of mild nonlinearities the linearization error of this approximation is acceptable. However, for this type of linearization the spread of \underline{x} , i.e., the covariance matrix \mathbf{C}^x is not taken into account and there is no measure which allows to quantify the linearization error.

B. Statistical Linear Regression

To overcome these flaws, deterministic sampling techniques are employed instead, which allow for propagating the mean and the covariance of \underline{x} through the nonlinear transformation (8). In doing so, linearizing the transformation by so-called statistical linear regression or *statistical linearization* is possible [10], [11]. More precisely, statistical linearization calculates a matrix $\underline{\mathbf{G}}$ and a vector \underline{b} such that

$$\underline{y} = \underline{g}(\underline{x}) \approx \underline{\mathbf{G}} \cdot \underline{x} + \underline{b}, \quad (9)$$

where the error term

$$\underline{e} = \underline{g}(\underline{x}) - \underline{\mathbf{G}} \cdot \underline{x} + \underline{b} \quad (10)$$

describes the deviation of the nonlinear transformation and its linear approximation. To determine $\underline{\mathbf{G}}$ and \underline{b} , the nonlinear

transformation $g(\cdot)$ is evaluated at a set of weighted *regression points* $\{\alpha_i, \underline{x}_i\}_{i=1 \dots L}$ with non-negative weights α_i with $\sum_i \alpha_i = 1$, which results in points $\underline{y}_i = g(\underline{x}_i)$ for $i = 1 \dots L$. This set of points is chosen in such a way that the mean $\hat{\underline{x}}$ and covariance \mathbf{C}^x of \underline{x} are captured exactly, that is

$$\hat{\underline{x}} = \sum_{i=1}^L \alpha_i \cdot \underline{x}_i \quad \text{and} \quad \mathbf{C}^x = \sum_{i=1}^L \alpha_i \cdot (\underline{x}_i - \hat{\underline{x}}) \cdot (\underline{x}_i - \hat{\underline{x}})^T.$$

Then \mathbf{G} and \underline{b} are determined by minimizing the weighted sum of squared errors

$$\{\mathbf{G}, \underline{b}\} = \arg \min_{\mathbf{G}, \underline{b}} \left(\sum_{i=1}^L \alpha_i \cdot \underline{e}_i^T \cdot \underline{e}_i \right) \quad (11)$$

with $\underline{e}_i = \underline{y}_i - (\mathbf{G} \cdot \underline{x}_i + \underline{b})$. The solution of (11) is given by

$$\mathbf{G} = (\mathbf{C}^{xy})^T (\mathbf{C}^x)^{-1} \quad \text{and} \quad \underline{b} = \hat{\underline{y}} - \mathbf{G} \cdot \hat{\underline{x}}, \quad (12)$$

where the set of propagated points $\{\alpha_i, \underline{y}_i\}_{i=1 \dots L}$ is used to approximate the mean, covariance, and cross-covariance of \underline{y} according to

$$\begin{aligned} \hat{\underline{y}} &\approx \sum_{i=1}^L \alpha_i \cdot \underline{y}_i, \quad \mathbf{C}^y \approx \sum_{i=1}^L \alpha_i \cdot (\underline{y}_i - \hat{\underline{y}}) \cdot (\underline{y}_i - \hat{\underline{y}})^T, \\ \mathbf{C}^{xy} &\approx \sum_{i=1}^L \alpha_i \cdot (\underline{x}_i - \hat{\underline{x}}) \cdot (\underline{y}_i - \hat{\underline{y}})^T. \end{aligned}$$

The linearization error is characterized by the error term (10) and has zero-mean and the covariance matrix

$$\mathbf{C}^e = \mathbf{C}^y - \mathbf{G} \mathbf{C}^x \mathbf{G}^T. \quad (13)$$

Thus, by means of the covariance \mathbf{C}^e it is possible to quantify the linearization error. If \mathbf{C}^e is a zero matrix, the density of the error \underline{e} corresponds to a Dirac delta distribution [12] and the transformation $g(\cdot)$ is affine with $g(\underline{x}) = \mathbf{G} \cdot \underline{x} + \underline{b}$.

C. Calculating the Regression Points

Many approaches for calculating the set of regression points have been proposed in the recent years. They differ in the number of regression points L and the way these points are chosen. In the following example, both selection schemes used in this paper are briefly introduced.

Example 1 In the simulations described in Sec. VI the famous *unscented transform* [5] and the *Gaussian estimator* [6] are considered. For both, the calculation of the sigma points $\underline{x}_i \in \mathbb{R}^{n_x}$ can be summarized as

$$\begin{aligned} \underline{x}_1 &= \hat{\underline{x}}, \\ \underline{x}_i &= \hat{\underline{x}} + \nu_j \cdot \mathbf{P}_l, \quad i = l + 1 + (j - 1) \cdot n_x, \end{aligned}$$

where \mathbf{P}_l , $l = 1 \dots n_x$ is the l th column of the matrix $\mathbf{P} = \sqrt{\mathbf{C}^x}$ and ν_j , $j = 1 \dots N$ are scaling factors. This results in a number of $L = n_x \cdot N + 1$ regression points. The type of the matrix root, the scaling factors, and the weights α_i of the regression points depend on the considered selection scheme.

In case of the unscented transform, the Cholesky decomposition is chosen as matrix root. For the scaling factors holds $N = 2$ and $\nu_1 = \sqrt{n_x + \kappa} = -\nu_2$, where κ is a scaling parameter. The weights are $\alpha_1 = \frac{\kappa}{n_x + \kappa}$ and $\alpha_i = \frac{1}{2(n_x + \kappa)}$, for $i > 1$.

The Gaussian estimator utilizes the eigenvalue decomposition for calculating the matrix root. The number of scaling factors N and thus the total number L of regression points can be varied. Since the scaling factors result from solving a optimization problem, there is no closed-form expression. For $N = 2$ and $N = 4$, the scaling factors can be calculated to

$$\begin{aligned} N = 2: \quad & \nu_j \in \{-1.2245, 1.2245\}, \\ N = 4: \quad & \nu_j \in \{-1.4795, -0.5578, 0.5578, 1.4795\}. \end{aligned} \quad (14)$$

The regression points are equally weighted with $\forall i: \alpha_i = \frac{1}{L}$.

It is important to note that the proposed adaptive Gaussian mixture filter is not restricted to these two selection schemes. In fact, any selection scheme for statistical linearization including those described in [13]–[15] can be used, depending on the considered application as well as the desired estimation performance and computational demand.

IV. SPLITTING SCHEME

Given a random vector \underline{x} , whose density function $f^x(\underline{x})$ is a Gaussian mixture with L^x components according to (6), it is possible to linearize the nonlinear transformation $g(\cdot)$ for each component of $f^x(\underline{x})$. This kind of component-wise or *local linearization* leads to an improved approximation of the true density function $f^y(\underline{y})$ of \underline{y} compared to a single, global linearization. To further improve the approximation, especially in case of strong nonlinearities and/or large variances of some components, the idea is to select a component of $f^x(\underline{x})$ and split it into several components with reduced weights and covariances. It was demonstrated for example in [8] that the filtering accuracy of local linearization approaches benefits from this decrease of the covariances and simultaneous increase of the number of Gaussians.

A. Component Selection

A straightforward way to select a Gaussian component for splitting is to consider the weights ω_i^x , $i = 1 \dots L^x$. The component with the highest weight is then split. This however does not take the nonlinearity of $g(\cdot)$ in the support of the selected component into account. Since linearization is performed component-wise and locally, a more reasonable selection would be to consider also the induced linearization error of each component. For this purpose, statistical linearization already provides an appropriate measure for the linearization error in form of the covariance matrix \mathbf{C}^e in (13).

In order to easily assess the linearization error in the multi-dimensional case, the trace operator is applied to \mathbf{C}^e , which gives the measure

$$\epsilon = \text{trace}(\mathbf{C}^e) \in [0, \infty). \quad (15)$$

Geometrically speaking, the trace is proportional to circumference of the covariance ellipsoid corresponding to \mathbf{C}^e . The larger \mathbf{C}^e and thus the linearization error, the larger is ϵ . Conversely, the trace is zero, if and only if \mathbf{C}^e is the zero matrix, i.e., $\epsilon = 0 \Leftrightarrow \mathbf{C}^e = \mathbf{0}$. Hence, (15) is only zero, when there is no linearization error, that is, the nonlinear transformation $g(\cdot)$ is affine in the support of the considered Gaussian component.

Besides the linearization error, the contribution of a component to the nonlinear transformation is important as well. That is, the probability mass of the component, which is given by its weight ω_i^x , has also to be taken into account. This avoids splitting irrelevant components. Putting all together the criterion for selecting a component i for splitting is defined as

$$s_i = (\omega_i^x)^\gamma \cdot (1 - \exp(-\epsilon_i))^{1-\gamma} \in [0, 1] \quad (16)$$

for $i = 1 \dots L^x$, where $1 - \exp(-\epsilon_i)$ normalizes the linearization measure (15) into the interval $[0, 1]$. For a geometric interpolation between weight and linearization error of component i , the parameter $\gamma \in [0, 1]$ is used. With $\gamma = 0$, selecting a component for splitting only focuses on the linearization error, while $\gamma = 1$ considers the weight only.

Component selection criteria for splitting have also been proposed in [16], [17]. The criterion in [16] is designed for the unscented transform only, while the criterion in [17] can only be calculated analytically in some special cases. The proposed criterion instead is generally applicable.

B. Splitting a Gaussian

Assume that according to the selection criterion (16), the Gaussian component $\omega \cdot \mathcal{N}(\underline{x}; \hat{\underline{x}}, \mathbf{C}^x)$ is chosen. Splitting this Gaussian into many can be formulated as replacing the Gaussian by a Gaussian mixture according to

$$\omega \cdot \mathcal{N}(\underline{x}; \hat{\underline{x}}, \mathbf{C}^x) \approx \sum_{j=1}^L \omega_j \cdot \mathcal{N}(\underline{x}; \hat{\underline{x}}_j, \mathbf{C}_j) . \quad (17)$$

It can be easily verified that for $L > 1$, the number of free parameters, i.e., weights, mean vectors and covariance matrices, is larger than the number of given parameters. More precisely, splitting a Gaussian is an ill-posed problem. In order to reduce the degrees of freedom and to not introduce errors concerning the mean and covariance, splitting is performed in a moment-preserving fashion. Thus, it must hold that

$$\begin{aligned} \omega &= \sum_{j=1}^L \omega_j \quad , \quad \hat{\underline{x}} = \sum_{j=1}^L \frac{\omega_j}{\omega} \cdot \hat{\underline{x}}_j \quad , \\ \mathbf{C}^x &= \sum_{j=1}^L \frac{\omega_j}{\omega} \cdot \left(\mathbf{C}_j + \hat{\underline{x}}_j \hat{\underline{x}}_j^T \right) - \hat{\underline{x}} \hat{\underline{x}}^T . \end{aligned} \quad (18)$$

To further simplify the problem, splitting is restricted in direction of the eigenvectors of \mathbf{C}^x , which is computationally cheap and numerically stable. Furthermore, it reduces the problem to splitting a univariate standard Gaussian.

1) *Univariate standard Gaussian*: A moment-preserving split of a univariate standard Gaussian $\mathcal{N}(x; 0, 1)$ into a mixture with L components requires to determine $3 \cdot L$ free parameters. By forcing symmetry, i.e., the means \hat{x}_j are placed symmetrically around the mean \hat{x} with symmetrically chosen weights and for the variances holds $\forall j : \sigma_j^2 = \sigma$, the number of free parameters is reduced to $L + 1$. In [18], a splitting library with symmetric components is proposed. Unfortunately, preserving the moments is not guaranteed. Instead, the following split into two components is used throughout this paper.

Example 2 Following the approach proposed in [19], the univariate standard Gaussian is split into the mixture $\frac{1}{2} \cdot \mathcal{N}(x; \hat{x}, \sigma^2) + \frac{1}{2} \cdot \mathcal{N}(x; -\hat{x}, \sigma^2)$. The moment-preserving constraints of splitting (18) lead to the dependency $\sigma^2 = 1 - \hat{x}^2$ between \hat{x} and σ , where \hat{x} is now the only free parameter. This equation is valid for $\hat{x} \in [-1, 1]$ and contains the trivial solution $\hat{x} = 0$. Generally, \hat{x} may be determined dynamically by minimizing the resulting linearization error. But throughout this paper, \hat{x} is set to 0.5 for simplicity.

To determine the parameters of more than two components, additional constraints, e.g., capturing higher order moments like the skewness or the kurtosis have to be considered additionally. Since splitting is performed recursively in this paper (see Sec. V), the new introduced components can be split in the subsequent splitting step if the local linearization error may not be reduced sufficiently. Splitting into two components is a good compromise between reducing the linearization error on the one hand and controlling the growth of the number of components and the computational load on the other hand.

2) *Multivariate Gaussian*: Applying univariate splitting to the multivariate case requires the eigenvalue decomposition of the covariance matrix $\mathbf{C}^x = \mathbf{V} \mathbf{D} \mathbf{V}^T$, with \mathbf{V} being the matrix of eigenvectors and \mathbf{D} being the diagonal matrix of eigenvalues according to

$$\mathbf{V} = [\underline{v}_1 \quad \underline{v}_2 \quad \dots \quad \underline{v}_{n_x}] \quad , \quad \mathbf{D} = \text{diag}(\lambda_1, \lambda_2, \dots, \lambda_{n_x}) \quad ,$$

where $\underline{v}_i \in \mathbb{R}^{n_x}$ are the (orthonormal) eigenvectors and λ_i are the eigenvalues. \mathbf{V} is a rotation matrix and the eigenvalue decomposition of \mathbf{C}^x corresponds to the transformation

$$\underline{x} = \mathbf{V} \cdot \underline{z} \quad (19)$$

of a Gaussian random vector \underline{z} with density

$$f^z(\underline{z}) = \mathcal{N}(\underline{z}; \hat{\underline{z}}, \mathbf{D}) = \prod_{i=1}^{n_x} \mathcal{N}(z_i; \hat{z}_i, \lambda_i) \quad (20)$$

to a Gaussian random vector \underline{x} with density $\mathcal{N}(\underline{x}; \hat{\underline{x}}, \mathbf{C}^x)$. Since the Gaussian $f^z(\underline{z})$ has a diagonal covariance matrix, the eigenvectors are parallel to the axes of the coordinate system. Thus, univariate splitting can be easily applied along the eigenvectors by replacing a univariate Gaussian on the right-hand side of (20) by a Gaussian mixture.

Assuming that eigenvector \underline{v}_l is chosen for splitting and let $\sum_{j=1}^L \omega'_j \cdot \mathcal{N}(z_l; \hat{z}'_j, \sigma_j^2)$ be the Gaussian mixture that approximates a univariate standard Gaussian as described above. As this mixture approximates a standard Gaussian, its components have to be shifted by adding \hat{z}_l and scaled by multiplying with $\sqrt{\lambda_l}$ in order to match the mean \hat{z}_l and the variance λ_l , respectively. These operations result in

$$\mathcal{N}(z_l; \hat{z}_l, \lambda_l) \approx \sum_{j=1}^L \omega'_j \cdot \mathcal{N}\left(z_l; \hat{z}_l + \sqrt{\lambda_l} \hat{z}'_j, \lambda_l \sigma_j^2\right) . \quad (21)$$

Plugging (21) into (20) leads to

$$\begin{aligned} \mathcal{N}(\underline{z}; \hat{\underline{z}}, \mathbf{D}) &\approx \\ &\sum_{j=1}^L \omega'_j \cdot \mathcal{N}\left(z_l; \hat{z}_l + \sqrt{\lambda_l} \hat{z}'_j, \lambda_l \sigma_j^2\right) \cdot \prod_{\substack{i=1 \\ i \neq l}}^{n_x} \mathcal{N}(z_i; \hat{z}_i, \lambda_i) . \end{aligned}$$

Transforming this mixture via (19) gives the desired splitting result (17) with the weights, means, and covariance matrices

$$\begin{aligned}\omega_j &= \omega \cdot \omega'_j, \\ \hat{\mathbf{x}}_j &= \hat{\mathbf{x}} + \sqrt{\lambda_l} \cdot \hat{\mathbf{z}}'_j \cdot \mathbf{v}_l, \\ \mathbf{C}_j &= \mathbf{C}^x + \lambda_l \cdot (\sigma_j^2 - 1) \cdot \mathbf{v}_l \mathbf{v}_l^T,\end{aligned}\quad (22)$$

for $j = 1 \dots L$.

It is worth mentioning that the calculation of the parameters in (22) is independent of the number of components L and does not necessarily require a symmetric, moment-preserving splitting. Thus, arbitrary splitting methods of univariate standard Gaussians besides those described in this paper, can be used with these formulae.

C. Splitting Direction

So far, no criterion for selecting an appropriate eigenvector for splitting is defined. A straightforward criterion may be the eigenvector with the largest eigenvalue as in [18], [19]. But since (16) determines the Gaussian component that causes the largest linearization error, merely splitting along the eigenvector with the largest eigenvalue does not take this error into account.

The key idea of the proposed criterion is to evaluate the deviation between the nonlinear transformation (8) and its linearized version (9) along each eigenvector. The eigenvector with the largest deviation is then considered for splitting, i.e., the Gaussian is split in direction of the largest deviation in order to cover this direction with more Gaussians, which will reduce the error in subsequent linearization steps.

By means of the error term (10), the desired criterion for the splitting direction is defined as

$$d_l := \int_{\mathbb{R}} \underline{\mathbf{e}}(\mathbf{x}_l(\nu))^T \cdot \underline{\mathbf{e}}(\mathbf{x}_l(\nu)) \cdot \mathcal{N}(\mathbf{x}_l(\nu); \hat{\mathbf{x}}, \mathbf{C}^x) d\nu \quad (23)$$

with $\mathbf{x}_l(\nu) := \hat{\mathbf{x}} + \nu \cdot \mathbf{v}_l$, $l = 1 \dots n_x$, and \mathbf{v}_l being the l th eigenvector \mathbf{C}^x . The integral in (23) cumulates the squared deviations along the l th eigenvector under the consideration of the probability at each point $\mathbf{x}_l(\nu)$. The eigenvector that maximizes (23) is then chosen for splitting. Unfortunately, due to the nonlinear transformation $\underline{\mathbf{g}}(\cdot)$ in (10), this integral cannot be solved in closed-form in general. For an efficient and approximate solution, the regression point calculation schemes described in Sec. III-C are employed to approximate the Gaussian in (23) in direction of \mathbf{v}_l by means of a Dirac mixture. This automatically leads to a discretization of the integral at a few but carefully chosen points.

V. ADAPTIVE GAUSSIAN MIXTURE FILTER

Based on the statistical linearization described in Sec. III and the splitting procedure proposed in Sec. IV, the complete adaptive Gaussian mixture filter (AGMF) is now derived. The key idea of AGMF is to dynamically increase the number of Gaussians of a given mixture at regions with large linearization errors. The number is reduced after each prediction and filtering in order to limit the computational and memory demand.

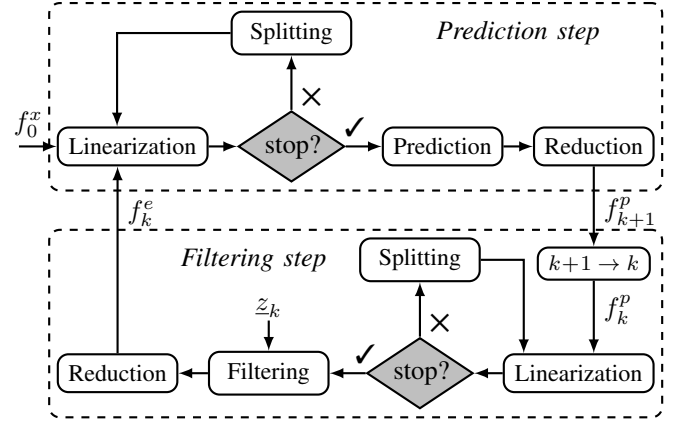


Figure 1. Flow chart of the proposed adaptive Gaussian mixture filter. Both the prediction and filtering step employ splitting and reduction for adapting the number of mixture components.

A. Prediction Step

The major operations to be performed in the prediction step are illustrated in the upper part of Fig. 1. The following paragraphs provide detailed descriptions of these operations.

1) *Linearization*: As shown in (7) the prediction step can be performed component-wise. Therefore, the nonlinear system function is linearized statistically at the weighted joint Gaussian $\omega_{k,s} \cdot \mathcal{N}(\mathbf{X}_k; \hat{\mathbf{x}}_{k,s}, \mathbf{C}_{k,s}^X)$ with $\mathbf{X}_k = [\mathbf{x}_k^T, \mathbf{w}_k^T]^T$, $s = (i-1) \cdot L_k^w + j = 1 \dots L_k^e \cdot L_k^w$, $\omega_{k,s} = \omega_{k,i}^e \cdot \omega_{k,j}^w$, and

$$\hat{\mathbf{x}}_{k,s} = \begin{bmatrix} \hat{\mathbf{x}}_{k,i}^e \\ \hat{\mathbf{w}}_{k,j}^w \end{bmatrix}, \quad \mathbf{C}_{k,s}^X = \begin{bmatrix} \mathbf{C}_{k,i}^e & \mathbf{0} \\ \mathbf{0} & \mathbf{C}_{k,j}^w \end{bmatrix}.$$

With (12), the linearization results in

$$\mathbf{x}_{k+1} = \underbrace{[\mathbf{A}_{k,s} \quad \mathbf{B}_{k,s}]}_{=\mathbf{G}_{k,s}} \cdot \mathbf{X}_k + \mathbf{b}_{k,s}. \quad (24)$$

2) *Splitting*: Due to the nonlinearity of the system function $\underline{\mathbf{a}}_k(\cdot)$, some of the mixture components $\omega_{k,s} \cdot \mathcal{N}(\mathbf{X}_k; \hat{\mathbf{x}}_{k,s}, \mathbf{C}_{k,s}^X)$ may locally cause severe linearization errors. These errors are quantified by means of the selection criterion (16). The component maximizing (16) will be split in direction of the largest deviation between the nonlinear function $\underline{\mathbf{a}}_k(\cdot)$ and its linearized version (24) as described in Sec. IV-C. After splitting this Gaussian, linearization is performed for the newly introduced mixture components. The linearization need not to be repeated for the remaining mixture components as they are not affected by the splitting.

Splitting Gaussians and the subsequent linearization is repeated until a stopping criterion is satisfied. This stopping criterion combines three user-defined thresholds:

- 1) For *each* component the value of the selection criterion (16) shall drop below the *error threshold* $\epsilon_{\max} \in [0, 1]$.
- 2) The number of Gaussians shall not grow beyond the *component threshold* L_{\max} .
- 3) The deviation between the original Gaussian mixture $\underline{\mathbf{f}}(\mathbf{x})$ and the mixture obtained via splitting $\hat{\underline{\mathbf{f}}}(\mathbf{x})$ shall remain below a *deviation threshold* $d_{\max} \in [0, 1]$.

In the latter case, the deviation is determined by means of the normalized integral squared distance measure [20]

$$D(f(\underline{x}), \tilde{f}(\underline{x})) = \frac{\int (f(\underline{x}) - \tilde{f}(\underline{x}))^2 d\underline{x}}{\int f(\underline{x})^2 d\underline{x} + \int \tilde{f}(\underline{x})^2 d\underline{x}} \in [0, 1] .$$

Since splitting always introduces an approximation error to the original mixture $f(\underline{x})$, tracking the deviation during splitting and keeping the deviation below the threshold d_{\max} avoids that errors introduced by splitting neutralize the gain in linearization. Splitting stops, if at least one threshold is reached.

3) *Prediction*: Let $\omega_{k,\tilde{s}} \cdot \mathcal{N}(\underline{X}_k; \hat{\underline{X}}_{k,\tilde{s}}, \mathbf{C}_{k,\tilde{s}}^X)$ be the Gaussians resulting from the splitting step, with $\tilde{s} = 1 \dots \tilde{L}_k^p$ and $\tilde{L}_k^p \gg L_k^e \cdot L_k^w$. Based on these Gaussians and their corresponding locally linearized system models (24), the parameters of each component of the predicted Gaussian mixture $f_{k+1}^p(\underline{x}_{k+1})$ can be calculated by means of the Kalman predictor according to

$$\begin{aligned} \omega_{k+1,\tilde{s}}^p &= \omega_{k,\tilde{s}} , \\ \hat{\underline{x}}_{k+1,\tilde{s}}^p &= \mathbf{A}_{k,\tilde{s}} \cdot \hat{\underline{x}}_{k,\tilde{s}}^e + \mathbf{B}_{k,\tilde{s}} \cdot \hat{\underline{w}}_{k,\tilde{s}} + \underline{b}_{k,\tilde{s}} , \\ \mathbf{C}_{k+1,\tilde{s}}^p &= \mathbf{A}_{k,\tilde{s}} \mathbf{C}_{k,\tilde{s}}^e \mathbf{A}_{k,\tilde{s}}^T + \mathbf{B}_{k,\tilde{s}} \mathbf{C}_{k,\tilde{s}}^w \mathbf{B}_{k,\tilde{s}}^T + \mathbf{C}_{k,\tilde{s}} , \end{aligned}$$

where $\mathbf{C}_{k,\tilde{s}}$ is the linearization error covariance (13).

4) *Reduction*: The number of components \tilde{L}_k^p in $f_{k+1}^p(\underline{x}_{k+1})$ grows due to the multiplication of the Gaussian mixtures $f_k^e(\underline{x}_k)$ and $f_k^w(\underline{w}_k)$ for prediction and due to splitting. It is necessary to bound this growth in order to reduce the computational and memory demand of subsequent prediction and filtering steps. For this purpose, one can exploit the redundancy and similarity of Gaussian components. Furthermore, many components will have weights close to zero, thus they can be removed without introducing significant errors. To reduce a Gaussian mixture, many algorithms have been proposed in the recent years (see for example [20]–[23]). Most of these algorithms require a *reduction threshold* L_{k+1}^p —typically much smaller than L_{\max} —to which the number of components of the given Gaussian mixture has to be reduced. In the simulations, Runnalls’ reduction algorithm [21] is employed as it provides a good trade-off between computational demand and reduction errors.

With the reduction to L_{k+1}^p components, the calculation of the predicted Gaussian mixture $f_{k+1}^p(\underline{x}_{k+1})$ in (6) is finished.

B. Filtering Step

The operations to be performed for the filtering step are almost identical to the prediction step (see Fig. 1). Thus, only linearization and filtering are described in the following. Splitting and reduction coincide with the prediction step.

1) *Linearization*: Linearization and filtering are also performed component-wise. Let $\omega_{k,s} \cdot \mathcal{N}(\underline{X}_k; \hat{\underline{X}}_{k,s}, \mathbf{C}_{k,s}^X)$ be the joint Gaussian comprising the i th component of the predicted mixture $f_k^p(\underline{x}_k)$ and the j th component of the measurement noise mixture (4), where $s = (i-1) \cdot L_k^v + j = 1 \dots L_k^p \cdot L_k^v$. The corresponding linearized measurement model is

$$\underline{z}_k = [\mathbf{H}_{k,s} \quad \mathbf{D}_{k,s}] \cdot \underline{\mathbf{X}}_k + \underline{b}_{k,s} \quad (25)$$

with joint state $\underline{\mathbf{X}}_k = [\underline{x}_k^T, \underline{v}_k^T]^T$.

2) *Filtering*: Let $\omega_{k,\tilde{s}} \cdot \mathcal{N}(\underline{X}_k; \hat{\underline{X}}_{k,\tilde{s}}, \mathbf{C}_{k,\tilde{s}}^X)$ be the Gaussians resulting from splitting, with $\tilde{s} = 1 \dots \tilde{L}_k^e$ and $\tilde{L}_k^e \gg L_k^p \cdot L_k^v$. Given the current measurement value \underline{z}_k , the Kalman filter update equations applied on these Gaussians and their corresponding locally linearized measurement models (25) give rise to the parameters of each component of the posterior Gaussian mixture $f_k^e(\underline{x}_k)$

$$\begin{aligned} \omega_{k,\tilde{s}}^e &= c_k \cdot \omega_{k,\tilde{s}} \cdot \mathcal{N}(\underline{z}_k; \hat{\underline{z}}_{k,\tilde{s}}, \mathbf{S}_{k,\tilde{s}}) , \\ \hat{\underline{x}}_{k,\tilde{s}}^e &= \hat{\underline{x}}_{k,\tilde{s}}^p + \mathbf{K}_{k,\tilde{s}} (\underline{z}_k - \hat{\underline{z}}_{k,\tilde{s}}) , \\ \mathbf{C}_{k,\tilde{s}}^e &= \mathbf{C}_{k,\tilde{s}}^p - \mathbf{K}_{k,\tilde{s}} \mathbf{H}_{k,\tilde{s}} \mathbf{C}_{k,\tilde{s}}^p , \end{aligned} \quad (26)$$

with predicted measurement $\hat{\underline{z}}_{k,\tilde{s}} = \mathbf{H}_{k,\tilde{s}} \cdot \hat{\underline{x}}_{k,\tilde{s}}^p + \mathbf{D}_{k,\tilde{s}} \cdot \hat{\underline{v}}_{k,\tilde{s}} + \underline{b}_{k,\tilde{s}}$, Kalman gain $\mathbf{K}_{k,\tilde{s}} = \mathbf{C}_{k,\tilde{s}}^p \mathbf{H}_{k,\tilde{s}}^T \mathbf{S}_{k,\tilde{s}}^{-1}$, innovation covariance $\mathbf{S}_{k,\tilde{s}} = \mathbf{H}_{k,\tilde{s}} \mathbf{C}_{k,\tilde{s}}^p \mathbf{H}_{k,\tilde{s}}^T + \mathbf{D}_{k,\tilde{s}} \mathbf{C}_{k,\tilde{s}}^v \mathbf{D}_{k,\tilde{s}}^T + \mathbf{C}_{k,\tilde{s}}$, and $\mathbf{C}_{k,\tilde{s}}$ being the linearization error covariance (13). The calculation of the weight $\omega_{k,\tilde{s}}^e$ in (26) is adapted from [8], [9], where $c_k = 1 / \sum_{\tilde{s}} \omega_{k,\tilde{s}} \cdot \mathcal{N}(\underline{z}_k; \hat{\underline{z}}_{k,\tilde{s}}, \mathbf{S}_{k,\tilde{s}})$ is a normalization constant.

After the reduction to L_k^e components, the posterior Gaussian mixture $f_k^e(\underline{x}_k)$ in (6) is completely determined.

VI. SIMULATION RESULTS

Two numerical simulations are conducted in order to demonstrate the performance of the proposed AGMF.

A. Shape Approximation

In the first simulation, the nonlinear growth process

$$\underline{y} = g(\underline{x}) = \frac{\underline{\xi}}{2} + 5 \cdot \frac{\underline{\xi}}{1 + \underline{\xi}^2} + \underline{w}$$

adapted from [24] is considered, where $\underline{x} = [\underline{\xi}, \underline{w}]^T \sim f^x(\underline{x}) = \mathcal{N}(\underline{x}; [1, 0]^T, \mathbf{I}_2)$ with \mathbf{I}_n being the $n \times n$ identity matrix. To approximate the density of \underline{y} , the Gaussian $f^x(\underline{x})$ is split recursively into a Gaussian mixture, where the number of components is always doubled until a maximum of 64 components is reached. No mixture reduction and no thresholds ϵ_{\max} , d_{\max} are used. The Gaussian estimator with 4 scaling factors according to (14) is employed for statistical linearization. The true density of \underline{y} is calculated via numerical integration.

Two different values for the parameter γ of the selection criterion (16) are used: $\gamma = 0.5$, which makes no preference between the component weight and the linearization error and $\gamma = 1$, which considers the weight only. Furthermore, a rather simple selection criterion is considered for comparison, where selecting a Gaussian for splitting is based on the weights only (as it is the case for $\gamma = 1$), while the splitting is performed in direction of the eigenvector with the largest eigenvalue.

Table I
APPROXIMATION ERROR (KLD $\times 10$) FOR DIFFERENT SPLITTING SCHEMES AND NUMBERS OF COMPONENTS.

splitting scheme	number of Gaussians						
	1	2	4	8	16	32	64
max. eigenvalue	2.01	0.77	0.64	0.47	0.39	0.21	0.26
$\gamma = 1$	2.01	0.77	0.59	0.34	0.20	0.12	0.07
$\gamma = 0.5$	2.01	0.77	0.40	0.22	0.07	0.03	0.02

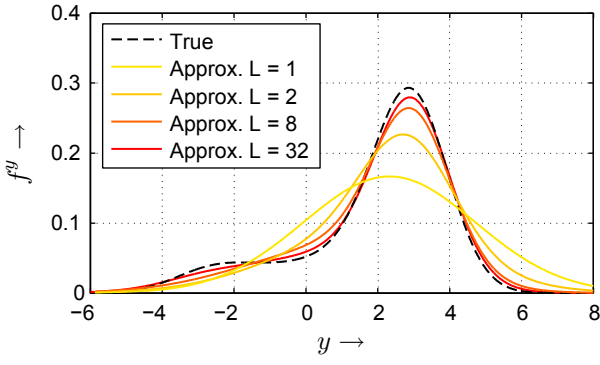


Figure 2. True density function of y (black, dashed) and approximations with an increasing number of mixture components.

Table I shows the Kullback-Leibler divergence (KLD, [25]) between the true density of y and the approximations obtained by splitting. The approximations of the proposed splitting scheme are significantly better than the approximations of the largest eigenvalue scheme. This follows from the fact that the proposed scheme not only considers the spread of a component. It also takes the linearization errors into account. In doing so, the Gaussians are always split along the eigenvector that is closest to ξ , since this variable is transformed nonlinearly, while w is not. This is different for the largest eigenvalue scheme, which wastes nearly half of the splits on w .

The inferior approximation quality for $\gamma = 1$ compared to $\gamma = 0.5$ results from splitting components, which may have a high importance due to their weight but which do not cause severe linearization errors. Thus, splitting these components will not improve the approximation quality much.

In Fig. 2, the approximate density of y is depicted for different numbers of mixture components for $\gamma = 0.5$. With an increasing number of components, the approximation approaches the true density very well.

B. Object Tracking

For the second simulation example, a object tracking scenario is considered. The kinematics of the mobile object are modeled by means of the bicycle model

$$\begin{bmatrix} \mathbf{x}_{k+1} \\ \mathbf{y}_{k+1} \\ \phi_{k+1} \end{bmatrix} = \begin{bmatrix} \mathbf{x}_k \\ \mathbf{y}_k \\ \phi_k \end{bmatrix} + \begin{bmatrix} \cos(\phi_k) \\ \sin(\phi_k) \\ u_k \end{bmatrix} + \mathbf{w}_k,$$

where the system state \mathbf{x}_k comprises the position $[\mathbf{x}_k, \mathbf{y}_k]^T$ and the orientation ϕ_k of the bicycle. At time step $k = 0$, the initial estimate of the state \mathbf{x}_0 is represented by a Gaussian density with mean $\hat{\mathbf{x}}_0 = [100 \text{ m}, 100 \text{ m}, 0 \text{ rad}]^T$ and covariance matrix $\mathbf{C}_0^x = \text{diag}([10^2, 10^2, \pi^2])$. The system input $u_k := \tan(\alpha_k)$ with α_k being the steering angle, is chosen randomly and uniformly distributed from the interval $[-0.2, 0.2]$ at each time step. The system noise \mathbf{w}_k is zero-mean Gaussian with covariance matrix $\mathbf{C}_k^w = \text{diag}(0.1^2, 0.1^2, 0.01^2)$.

A radar sensor with measurement model

$$\mathbf{z}_k = \begin{bmatrix} \sqrt{\mathbf{x}_k^2 + \mathbf{y}_k^2} \\ \arctan(\mathbf{y}_k/\mathbf{x}_k) \end{bmatrix} + \mathbf{v}_k$$

is employed for observing the object, where the measurement noise \mathbf{v}_k is modeled as unimodal glint noise [26] with density $f_k^v(\mathbf{v}_k) = (1 - \beta) \cdot \mathcal{N}(\mathbf{v}_k; \mathbf{0}, \mathbf{C}_{k,1}^v) + \beta \cdot \mathcal{N}(\mathbf{v}_k; \mathbf{0}, \mathbf{C}_{k,2}^v)$ with covariances $\mathbf{C}_{k,1}^v = \text{diag}(1^2, 0.1^2)$ and $\mathbf{C}_{k,2}^v = \text{diag}(2^2, 0.2^2)$. The parameter β refers to the glint noise probability. Six probability values $\beta = \{0, 0.2, \dots, 1\}$ are exploited for simulation. By increasing β it is possible to investigate the performance of the filters for stronger noise, which is also heavily tailed for $\beta \neq 0$ and $\beta \neq 1$.

For this simulation setup, AGMF is applied with parameter $\gamma = 0.5$, error threshold $\epsilon_{\max} = 0.05$, deviation threshold $d_{\max} = 1$, and component threshold $L_{\max} = 128$ for both prediction and filtering. Three values of reduction thresholds are used, $L_k^p = L_k^e = 2, 8, 32$. For comparison, a Gaussian mixture filter (denoted as MWE) employing the simple largest-weight-largest-eigenvalue-criterion as described in the previous section is considered. Further, the adaptive level of detail (ALD) Gaussian mixture filter proposed in [16] is employed as well. Since ALD is only designed for the unscented transform (see Example 1), this statistical linearization method is also used for AGMF to allow a fair comparison. The scaling parameter κ of the unscented transform is set to 0.5, i.e., all regression points are equally weighted. MWE and ALD use the same parameters as AGMF, except that MWE always splits until L_{\max} is reached since it exploits no linearization errors.

Besides these Gaussian mixture filters, a particle filter (PF) with residual resampling and 10,000 samples as well as the unscented Kalman filter (UKF, [4]) with $\kappa = 0.5$ are also applied.

For each glint probability and each reduction threshold, 50 Monte Carlo simulation runs are performed, where the object is observed for 100 time steps. In Fig. 3 (a), the average root means square error (rmse) of the position and the average runtime per simulation run are depicted. The AGMFs with 8 and 32 components provide the best tracking performance. The PF is close to AGMF, but with a significantly higher runtime. Conversely, the UKF is by far the fastest algorithm, but leads to diverging estimates.

The splitting criterion used for ALD selects components that exhibit a high degree of nonlinearity. But splitting is performed merely in direction of the largest eigenvalue. This explains the relative poor tracking performance of ALD.

Even if MWE is allowed to split until L_{\max} is reached, the performance of MWE is always inferior to AGMF. This is due to wasting many splits, e.g., in the prediction step only one quarter and less of the splits is used for ϕ_k , which is the only nonlinearly transformed variable. Here, AGMF is much more effective thanks to the novel splitting criterion. Besides splitting mainly in direction of the nonlinearity, it does not require all available splits as shown in Fig. 3 (b). The maximum number of splits is $L_{\max} - L_k^p$ in the prediction step and analogously in the filtering step. But at most 40 splits are performed in case of the strongest noise and when the state mixture is reduced to two components. If more components are allowed to represent the state density, the number of splits decreases as the approximation before splitting is already of

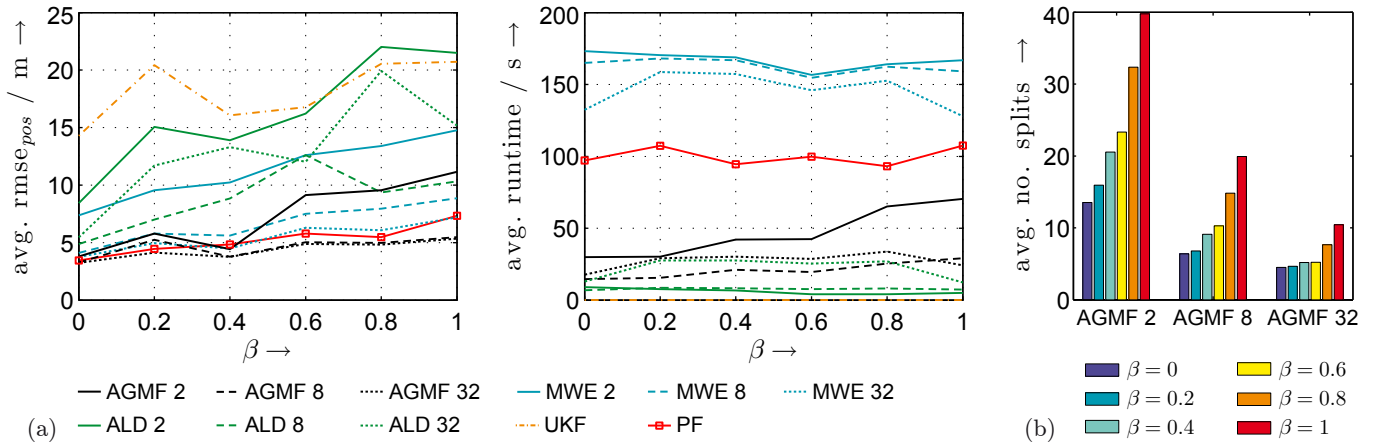


Figure 3. (a) Average rmse over all simulation runs and average runtime per simulation run for each β . (b) Average number of splits performed per prediction or filtering step of the AGMF for each β .

high quality. This also reduces the runtime as can be seen when comparing for example AGMF 32 with AGMF 2. Here, the time consuming splitting operation has to be performed less often and the reduction operation has to reduce a mixture with an already low number of components.

VII. CONCLUSIONS

In this paper, a novel adaptive Gaussian mixture filter has been proposed. It is based on statistical linearization, which allows quantifying the induced linearization errors in terms of a linearization error covariance matrix. A criterion based on this covariance matrix is used for selecting Gaussian components for splitting, while the direction of the split is performed in direction of the eigenvalue with the strongest linearization errors. Compared to other splitting criteria, the proposed one reliably detects strong nonlinearities and keeps the number of splits on a low level. Furthermore, arbitrary approaches for statistical linearization can be employed.

REFERENCES

- [1] M. S. Arulampalam, S. Maskell, N. Gordon, and T. Clapp, "A Tutorial on Particle Filters for Online Nonlinear/Non-Gaussian Bayesian Tracking," *IEEE Transactions on Signal Processing*, vol. 50, no. 2, pp. 174–188, Feb. 2002.
- [2] R. E. Kalman, "A new Approach to Linear Filtering and Prediction Problems," *Transactions of the ASME, Journal of Basic Engineering*, vol. 82 (Series D), no. 1, pp. 35–45, 1960.
- [3] D. Simon, *Optimal State Estimation: Kalman, H-Infinity, and Nonlinear Approaches*, 1st ed. Wiley & Sons, 2006.
- [4] E. A. Wan and R. van der Merwe, "The Unscented Kalman Filter for Nonlinear Estimation," in *Adaptive Systems for Signal Processing, Communications, and Control Symposium 2000*, 2000, pp. 153–158.
- [5] S. J. Julier and J. K. Uhlmann, "Unscented Filtering and Nonlinear Estimation," *Proceedings of the IEEE*, vol. 92, no. 3, pp. 401–422, 2004.
- [6] M. F. Huber and U. D. Hanebeck, "Gaussian Filter based on Deterministic Sampling for High Quality Nonlinear Estimation," in *Proceedings of the 17th IFAC World Congress*, Seoul, Republic of Korea, Jul. 2008.
- [7] V. Maz'ya and G. Schmidt, "On approximate approximations using gaussian kernels," *IMA J. Numer. Anal.*, vol. 16, pp. 13–29, 1996.
- [8] D. L. Alspach and H. W. Sorenson, "Nonlinear Bayesian Estimation using Gaussian Sum Approximation," *IEEE Transactions on Automatic Control*, vol. 17, no. 4, pp. 439–448, Aug. 1972.
- [9] M. Simandl and J. Dunik, "Sigma point gaussian sum filter design using square root unscented filters," in *Proceedings of the 16th IFAC World Congress*, Prague, Czech Republic, Jul. 2005.
- [10] T. Lefebvre, H. Bruyninckx, and J. D. Schutter, *Nonlinear Kalman Filtering for Force-Controlled Robot Tasks*. Springer Berlin, 2005.
- [11] T. Vercauteren and X. Wang, "Decentralized Sigma-Point Information Filters for Target Tracking in Collaborative Sensor Networks," *IEEE Transactions on Signal Processing*, vol. 53, no. 8, pp. 2997–3009, 2005.
- [12] K. R. Parthasarathy, *Probability Measures on Metric Spaces*, new ed. American Mathematical Society, 2005.
- [13] I. Arasaratnam and S. Haykin, "Cubature Kalman Filters," *IEEE Transactions on Automatic Control*, vol. 54, no. 6, pp. 1254–1269, Jun. 2009.
- [14] K. Ito and K. Xiong, "Gaussian Filters for Nonlinear Filtering Problems," *IEEE Transactions on Automatic Control*, vol. 45, no. 5, pp. 910–927, May 2000.
- [15] T. S. Schei, "A finite difference method for linearizing in nonlinear estimation algorithms," *Automatica*, vol. 33, no. 11, pp. 2051–2058, Nov. 1997.
- [16] F. Faubel and D. Klakow, "An Adaptive Level of Detail Approach to Nonlinear Estimation," in *Proceedings of the 2010 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2010, pp. 3958–3961.
- [17] A. Rauh, K. Briechele, and U. D. Hanebeck, "Nonlinear Measurement Update and Prediction: Prior Density Splitting Mixture Estimator," in *Proceedings of the 2009 IEEE International Conference on Control Applications (CCA)*, Jul. 2009.
- [18] M. F. Huber, T. Bailey, H. Durrant-Whyte, and U. D. Hanebeck, "On Entropy Approximation for Gaussian Mixture Random Vectors," in *Proceedings of the 2008 IEEE International Conference on Multisensor Fusion and Integration for Intelligent Systems (MFI)*, Seoul, Republic of Korea, Aug. 2008, pp. 181–188.
- [19] F. Faubel, J. McDonough, and D. Klakow, "The Split and Merge Unscented Gaussian Mixture Filter," *IEEE Signal Processing Letters*, vol. 16, no. 9, pp. 786–789, Sep. 2009.
- [20] M. F. Huber and U. D. Hanebeck, "Progressive Gaussian Mixture Reduction," in *Proceedings of the 11th International Conference on Information Fusion (Fusion)*, Cologne, Germany, Jul. 2008.
- [21] A. R. Runnalls, "Kullback-Leibler Approach to Gaussian Mixture Reduction," *IEEE Transactions on Aerospace and Electronic Systems*, vol. 43, no. 3, pp. 989–999, Jul. 2007.
- [22] D. J. Salmond, "Mixture reduction algorithms for target tracking in clutter," in *Proceedings of SPIE Signal and Data Processing of Small Targets*, vol. 1305, Oct. 1990, pp. 434–445.
- [23] M. West, "Approximating Posterior Distributions by Mixtures," *Journal of the Royal Statistical Society: Series B*, vol. 55, no. 2, pp. 409–422, 1993.
- [24] G. Kitagawa, "Monte Carlo Filter and Smoother for Non-Gaussian Non-linear State Space Models," *Journal of Computational and Graphical Statistics*, vol. 5, no. 1, pp. 1–25, 1996.
- [25] T. M. Cover and J. A. Thomas, *Elements of Information Theory*. John Wiley & Sons, Inc., 1991.
- [26] W.-R. Wu, "Target Tracking with Glint Noise," *IEEE Transactions on Aerospace and Electronic Systems*, vol. 29, no. 1, pp. 174–185, 1993.